Linux & HPC

(a personal view)

Marc Snir

Thomas J. Watson Research Center
PO Box 218
Yorktown Heights, NY 10598

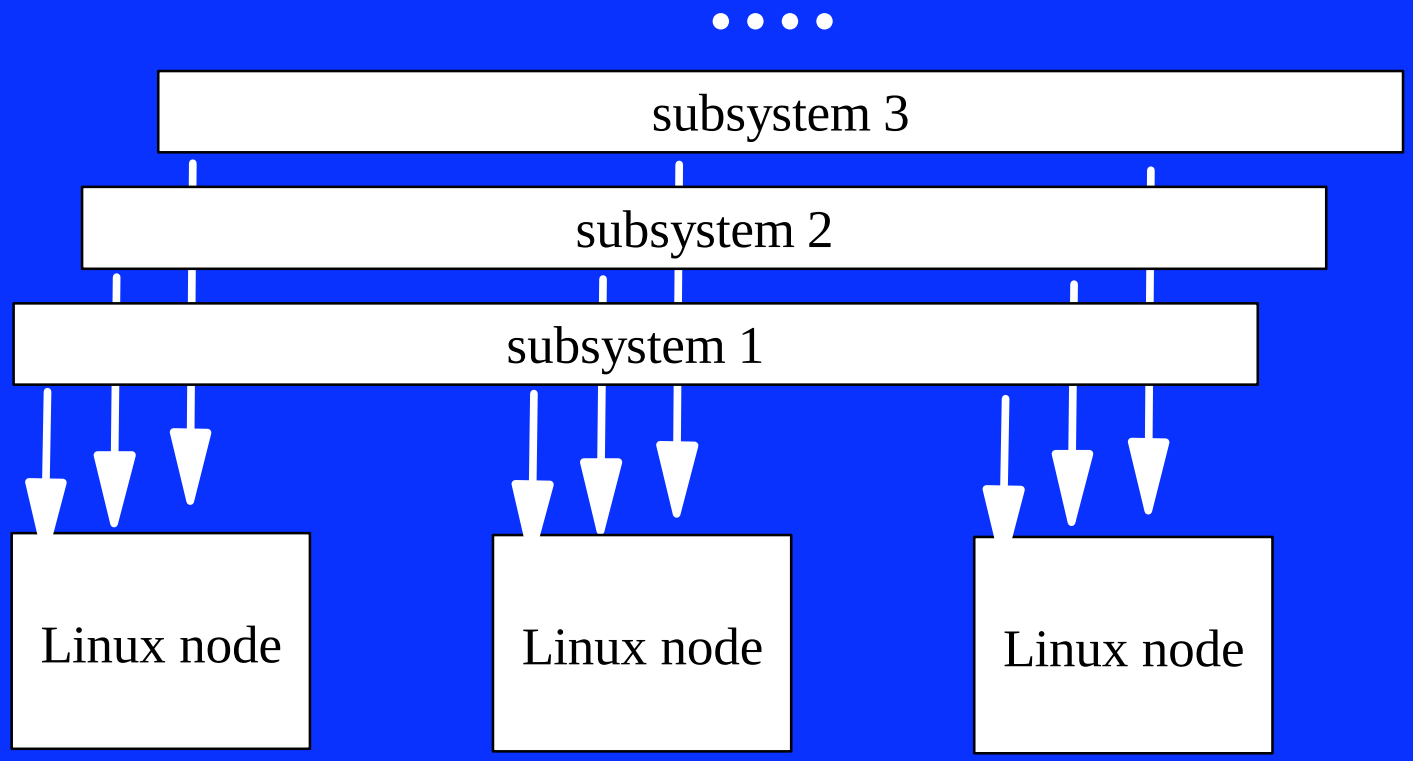
March 00

Scalable clusters of small SMP nodes -- sw stack

	scale independent (node software)	scale dependent (cluster software)
general	node OS, node C compiler, node debugger,...	cluster management: bringup, configuration management, system monitoring, logging, recovery,...
domain specific	Open MP compiler, Lapack,....	MPI, Scalapack, parallel job scheduler, PFS, checkpt/restart,...

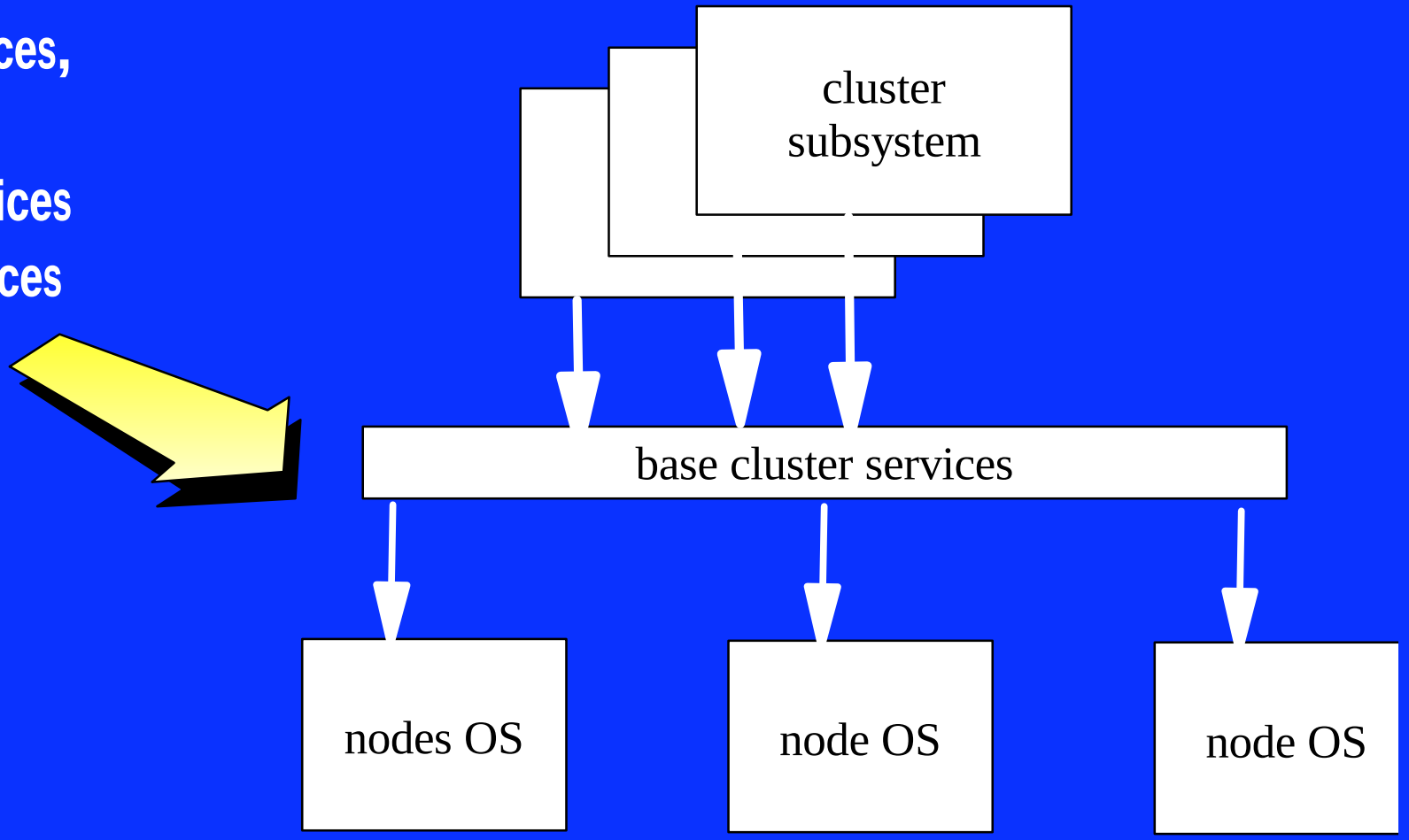
	scale independent	scale dependent
general	available "for free" in OSS, at 1st approx. (need improvements for storage servers, manager nodes,...)	DEVELOPED BY INDUSTRY TO TAKE CARE OF LARGE SERVER FARMS! (web/app hosting)
domain specific	Requires work (port), but no new invention. ISVs or HPC community?	MUST BE DONE BY HPC COMMUNITY!

How many daemons can stand on the top of a pin?



Toward a cluster OS

- heartbeat services,
- group services,
- messaging services
- broadcast services
- ...



This should be first / main focus of HPC community / industry OSS effort!

Know where you go -- Caveat OSS Emptor

- HPC OSS effort is not (in coming 2--3 years) about building a better sw environment than currently available on vendor platforms; it's about catching up.
- There is a lot of grunge work (test, tuning) having to do with large system effects (scalability); there will not be "thousands of watching eyes" to catch / fix these problems.
- Clusters are still hard to program, even if Linux driven. Efforts toward support of higher-level programming models are little helped by shift to OSS.
  - work in this domain already is OSS!

And what about large (NUMA) SSI SMPs?

- **Linux does not scale today**
- **Mainstream Linux effort does not lead to kernel that scales beyond modest size SMPs**
  - not main interest of Linux core team
  - cannot be handled by patches atop standard Linux distribution; needs significant restructuring of kernel (and separate distribution)
  - has a cost (code complexity and size, reduced small system performance) that may not be desireable for mainstream Linux
  - requires large / expensive platform to develop / test / tune
  - requires a lot of grunge work if based on current (one generation old) OS technologies
  - performs only when "OS is out of the way" most of the time.
    - important cause to primitive state of parallel programming environments!
- **Short term solution: Linux APIs / ABIs supported atop mature (vendor) kernel**
- **Long term solution: Start from scratch with new OS technology (e.g., K42)**